Machine Learning MC321
Lab Assignment 2
(Linear Regression)

January 25, 2025

Note:

1. Comment on the observation of your results.
2. This exercise is to solidify your understanding of univariate and multivariate linear regression through:
   - Analytical approach (closed-form solution or normal equation),
   - Numerical approach (gradient descent).

Datasets: Real Estate Dataset link

Question 1

Perform linear regression using the closed-form solution (normal equation).

$$\mathbf{w} = (X^T X)^{-1} X^T \mathbf{y}$$

where X is the $N \times (D+1)$ matrix of input features where N is the number of training examples and D is the size of the feature vector, y is the target vector of size $N \times 1$ such that $\mathbf{y}^{(i)}$ is the target variable corresponding to sample $\mathbf{x}^{(i)}$ and weight vector $\mathbf{w}$ is $(D+1) \times 1$. Note that the first column of X is all 1's corresponding to the intercept term.

(a) On the datasets provided.

(b) For the following X and $\mathbf{y}$, use scikit-learn and normal equation to learn a linear model. You may find that one of the matrices in the normal equation is non-invertible. Why does the matrix turn out to be non-invertible? Why can scikit-learn implementation still correctly solve this regression problem?

$$X = \begin{bmatrix} 1 & 2 \\ 2 & 4 \\ 3 & 6 \\ 4 & 8 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} 2 \\ 3 \\ 4 \\ 5 \end{bmatrix}$$

Question 2

Implement batch gradient descent, stochastic gradient descent and minibatch gradient descent for the function $\mathbf{f}(\mathbf{w}) = \sum_{i=1}^{m}(y_i - (w_0 + w_1 x_i))^2$ where m is the number of data samples in the dataset provided.

1. Plot the surface for the loss function $\mathbf{f}(\mathbf{w}) = \sum_{i=1}^{m}(y_i - (w_0 + w_1 x_i))^2$ to visualize its shape. Is it convex or non-convex? Verify it quantitatively.
2. Plot the loss function $\mathbf{f}(\mathbf{w}) = \sum_{i=1}^{m}(y_i - (w_0 + w_1 x_i))^2$ against iteration t (or every 10 iterations) for different combinations of initial guess and learning rate. For a function, you can superimpose the results in different settings. Use the legend to specify the initial guess and values for each setting
3. Create a Matplotlib animation of the contour plot. The different frames in the animation correspond to different iterations of gradient descent applied on the dataset to learn w_0 and w_1. For each iteration, draw the current value of $\mathbf{w}^{(t)}$ on the contour plot and also an arrow to the next $\mathbf{w}^{(t+1)}$ as learnt by gradient update rule. The overall title of the plot shows the iteration number and convergence. Use a legend to show the results of all three algorithms on a single plot. Comment on your observations.

Question 3

Try to fit a linear model using mean absolute error (MAE) function instead of least squares (MSE). Randomly for 10 different values of y add a positive or negative number to make the data points outliers. Which loss function is more robust to outliers visually as well as quantitatively based on residual error $|y_i - h_w(x_i)|$ and $(y_i - h_w(x_i))^2$ for MAE and MSE based loss functions, respectively?

$$J(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N} |\mathbf{y}^{(i)} - h_{\mathbf{w}}(\mathbf{x}^{(i)})|$$